

# CSS3 para Juniors

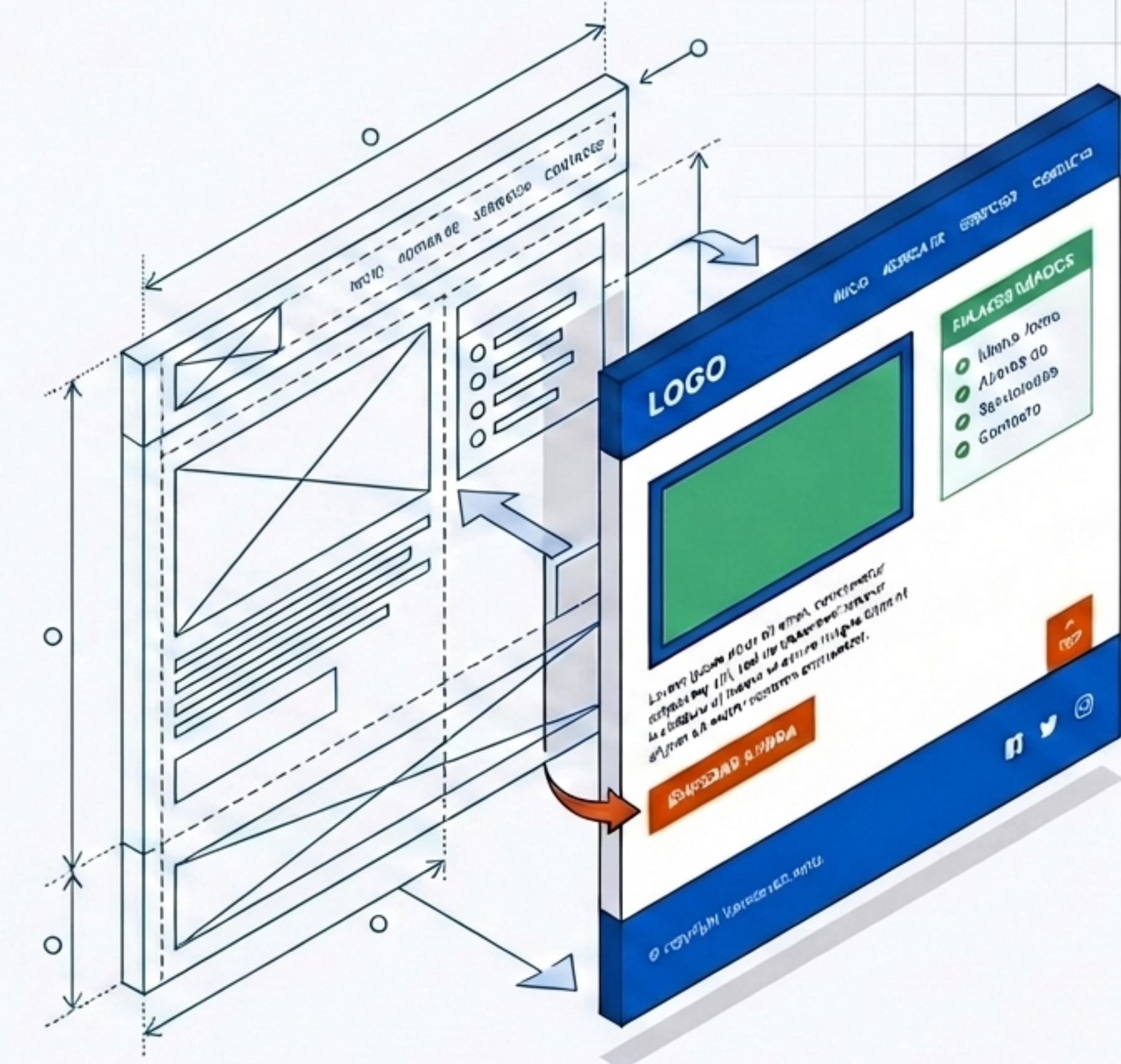
Diseño profesional y  
maquetación moderna

Guía clara para escribir CSS  
organizado, mantenable y profesional.

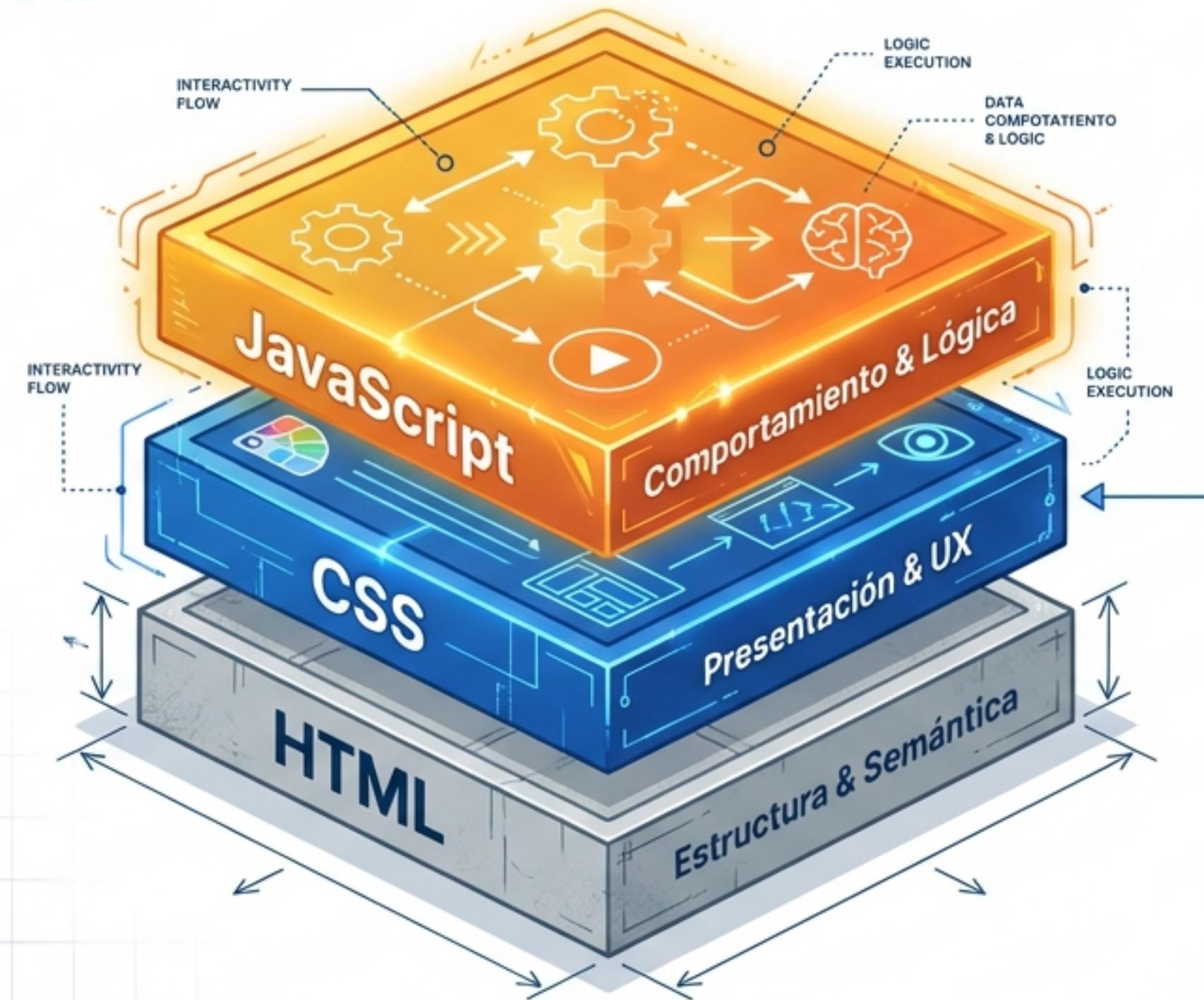
Autor: Vanesa Paola Soria

Marca: <https://conecta-jr.vercel.app/>

Contexto: Material exclusivo para  
desarrolladores.



# 1. Introducción: El Ecosistema del Desarrollo Web



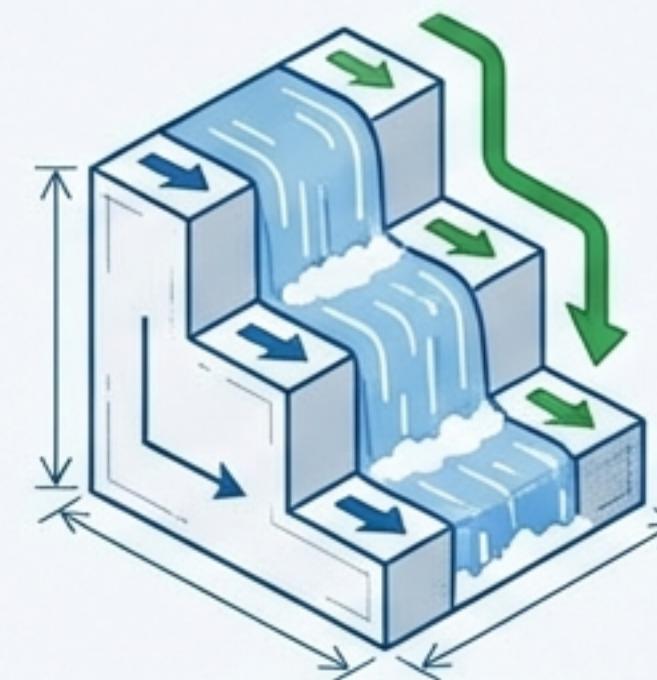
Un **CSS bien estructurado** no solo embellece; impacta directamente en la **escalabilidad del proyecto** y la **velocidad de carga**.

// Key Performance Indicator (KPI)

## 2. ¿Qué es CSS? La Lógica del Estilo

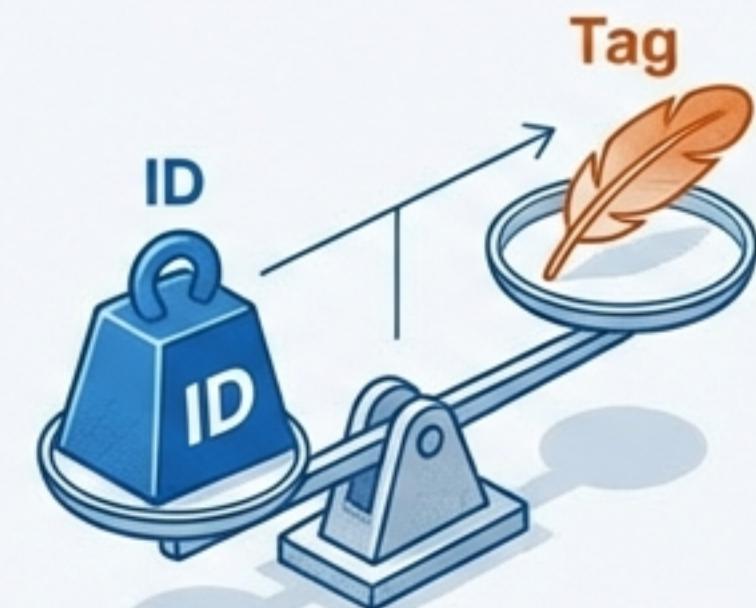
CSS (Hojas de Estilo en Cascada) define cómo se ve el documento.

### Cascada



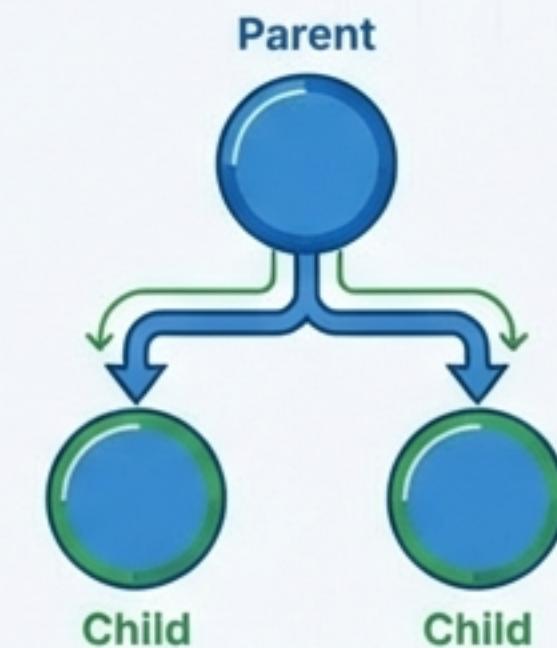
El orden importa. La última regla leída gana si la especificidad es igual.

### Especificidad



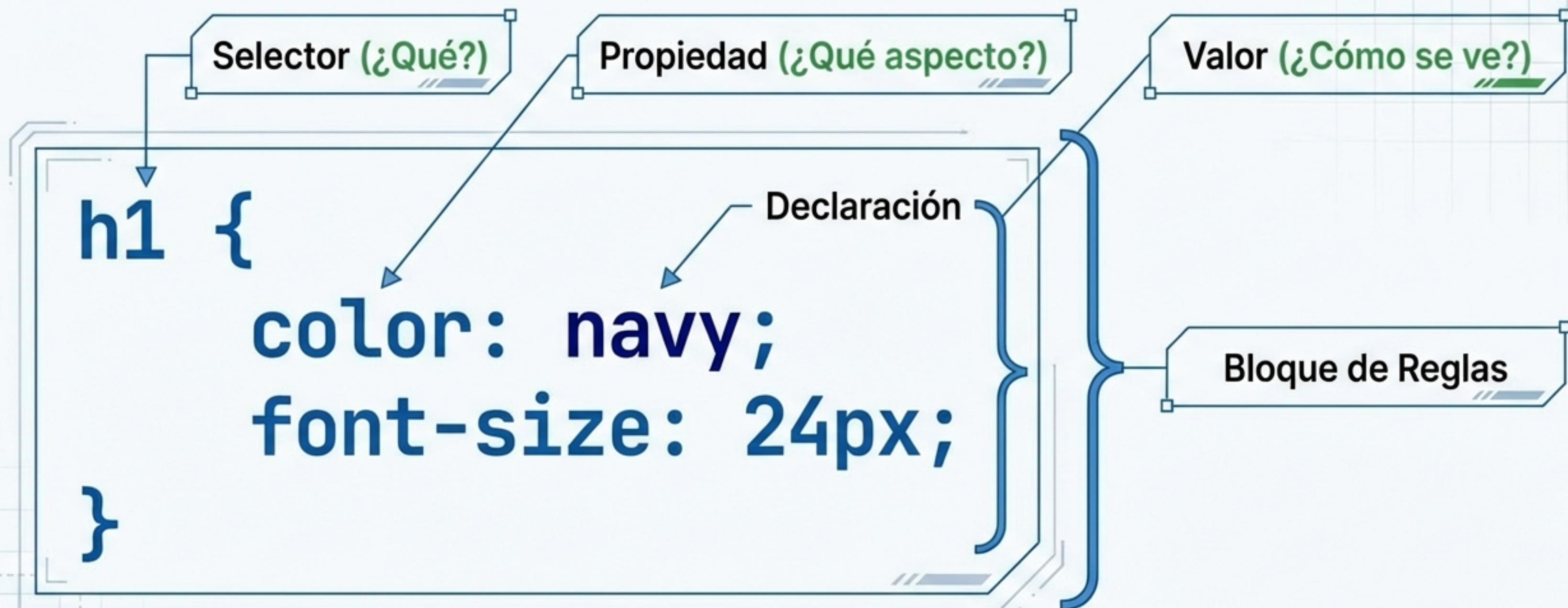
El 'peso' de una regla. Un ID pesa más que una Clase.

### Herencia



Propiedades que pasan de padres a hijos (ej: font-family).

### 3. Sintaxis Básica: Anatomía de una Regla

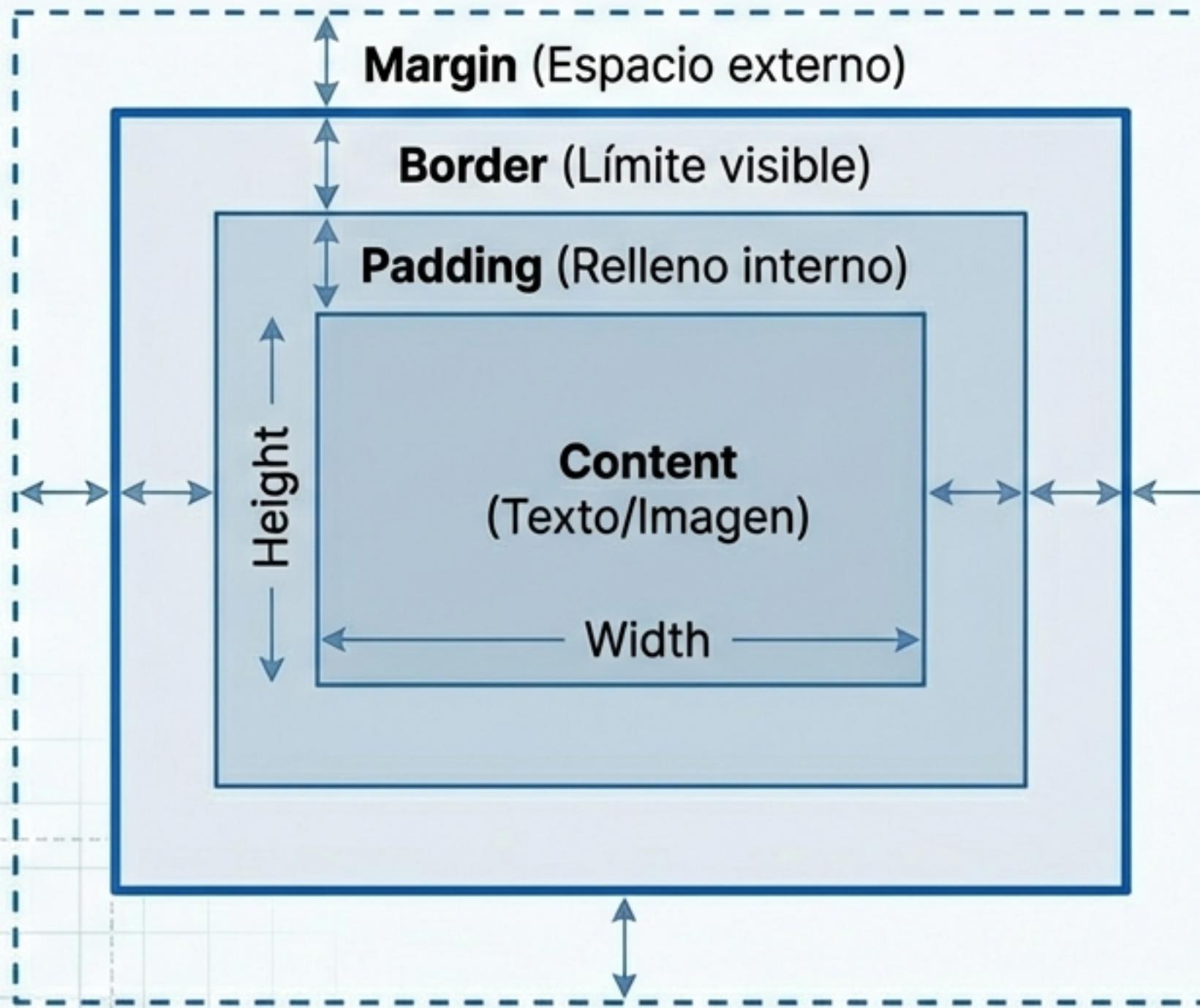


La puntuación es crítica: llaves para agrupar, dos puntos para asignar, punto y coma para finalizar.

## 4. Selectores Fundamentales

	Etiqueta	<code>div</code>	Baja especificidad. Afecta a todos los elementos del tipo.
	Clase	<code>.boton</code>	Especificidad media. Reutilizable. La herramienta principal.
	ID	<code>#header</code>	Alta especificidad. Único por página. Usar con precaución.
Descendiente ( <code>div p</code> ) - Elementos dentro de otros.		Combinado ( <code>.menu.activo</code> ) - Elementos con ambas clases.	

# 5. El Modelo de Caja (Box Model)



## Propiedad Crítica

`box-sizing: border-box;`  
Cambia el cálculo del ancho para incluir padding y border.  
Evita que los elementos rompan el layout inesperadamente.

# 6. Propiedades Esenciales

## Estética (Visual)



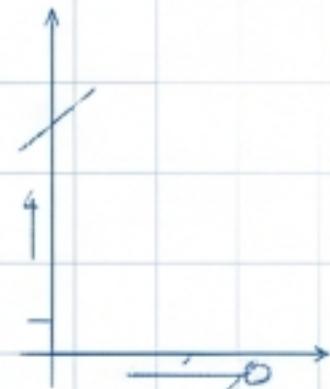
**color / background**: Define la paleta.



**font-family / font-size**: Define la tipografía.



**text-align**: Alineación del contenido.



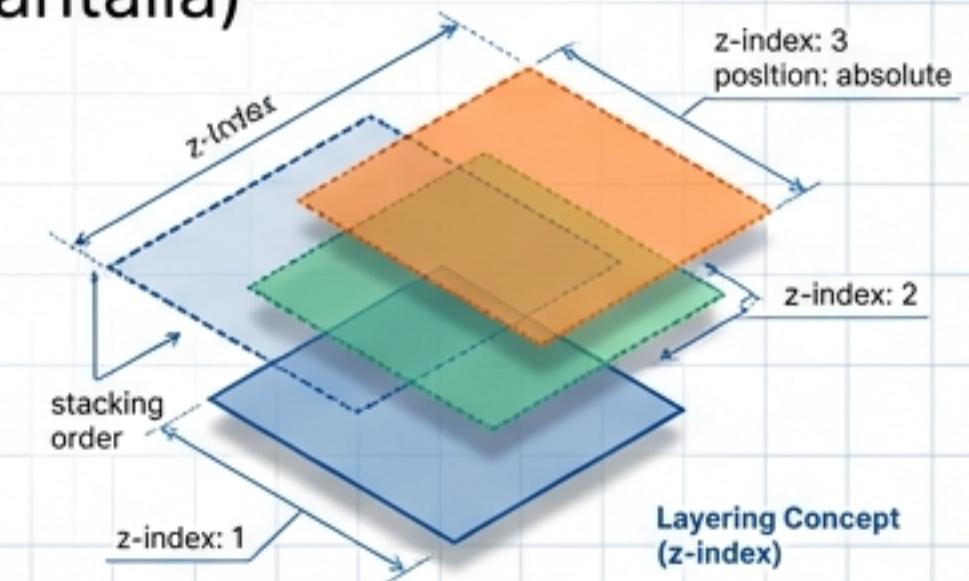
## Comportamiento (Estructura)

### display:

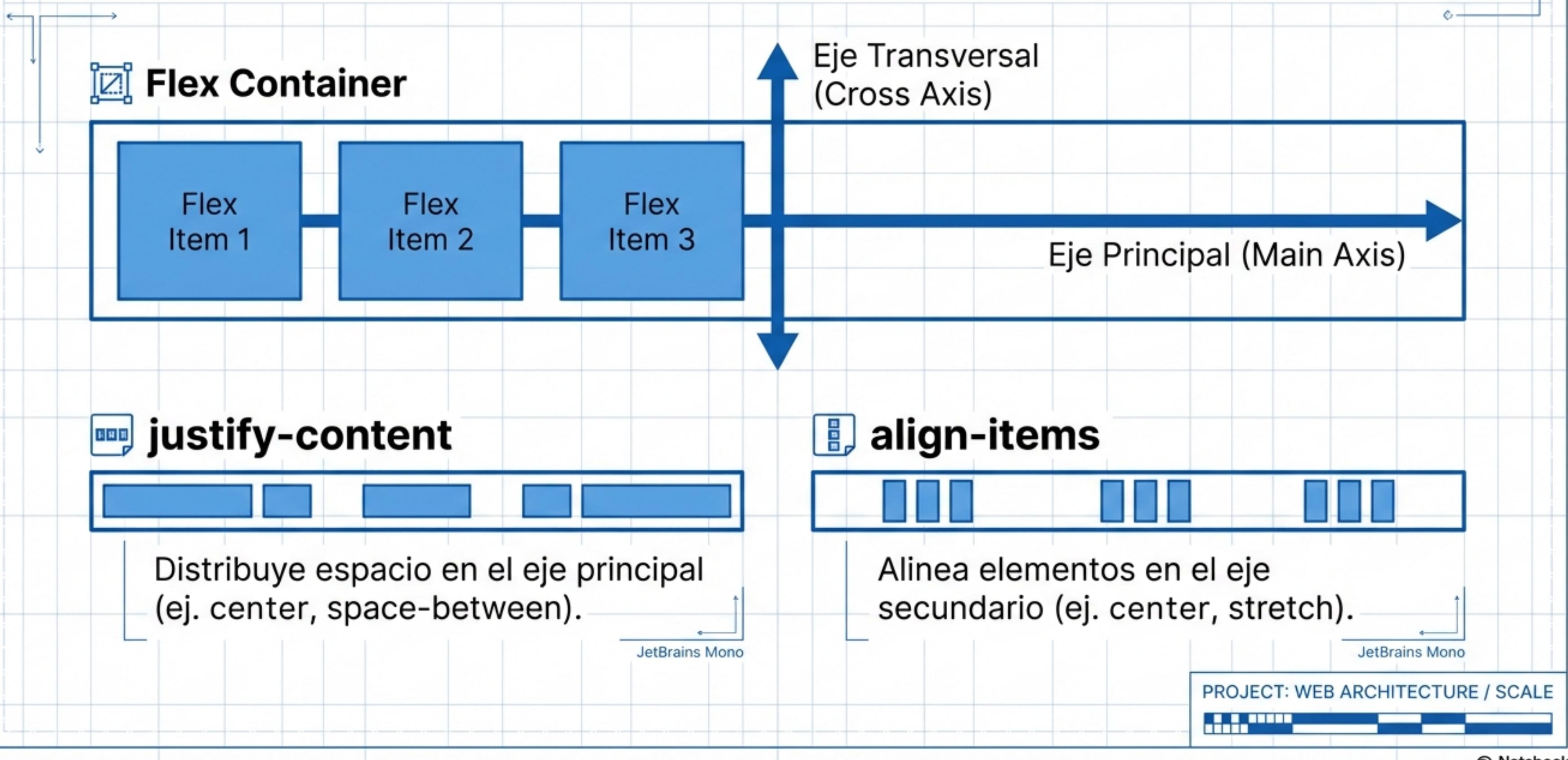
- **block** (ocupa todo el ancho)
- **inline** (fluye con texto)

### position:

- **static** (flujo normal)
- **absolute** (fuera del flujo)
- **fixed** (fijo en pantalla)



# 7. Layout Moderno: Flexbox (Arquitectura 1D)

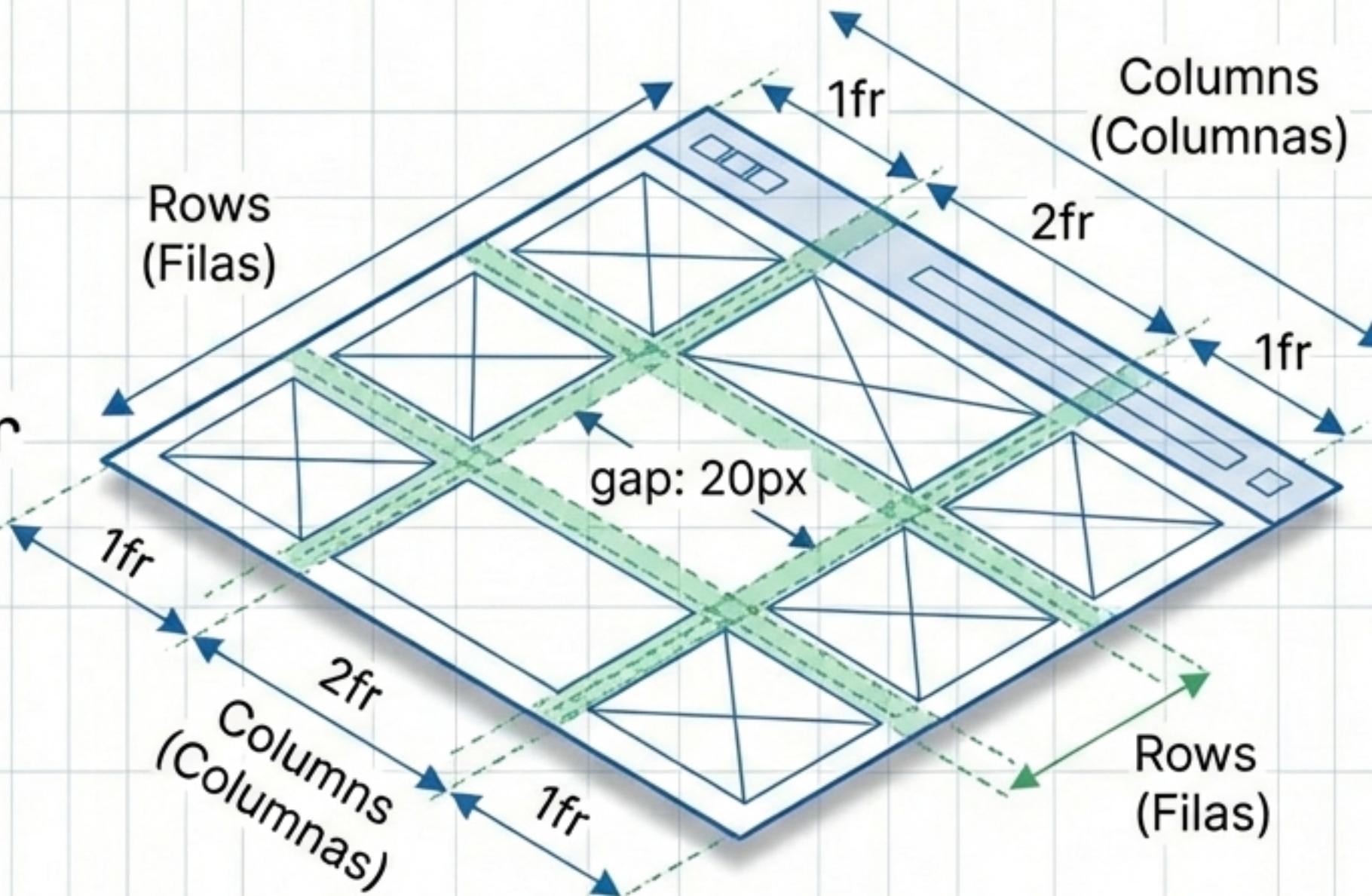


# 7. Layout Moderno: CSS Grid (Arquitectura 2D)

Grid permite diseñar en dos direcciones simultáneamente:  
Filas y Columnas.

- `display: grid;`
- `grid-template-columns: 1fr 2fr 1fr;` (Definición de columnas)
- `gap: 20px;` (Canales/Espacio entre celdas)

Ideal para el esqueleto general de la página.

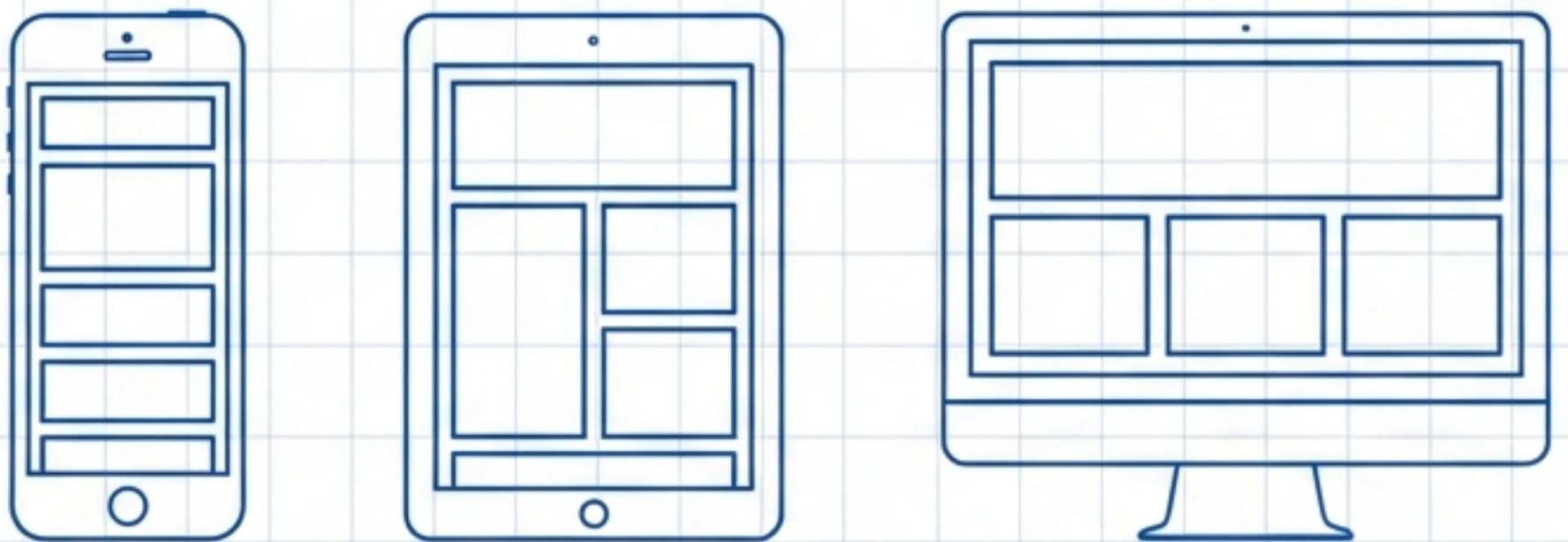


PROJECT: WEB ARCHITECTURE / SCALE



# 8. Responsive Design: Adaptabilidad

**Mobile First** (Diseñar de menor a mayor)



```
@media (min-width: 768px) {  
    .contenedor {  
        width: 100%;  
    }  
}
```

- **Media Queries:** Condicionales basados en el ancho de pantalla.
- **Unidades Relativas:** Usar %, rem, vw en lugar de px fijos.



# 9. Buenas Prácticas Profesionales



## Nomenclatura BEM

Bloque-Elemento--Modificador. Ej: .tarjeta\_\_titulo--grande



## Organización

Mantén archivos estructurados y comentados. Separación de preocupaciones.



## Evitar !important

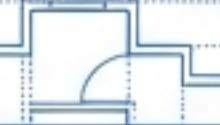
Rompe la cascada y dificulta el mantenimiento. Usar solo en casos extremos.



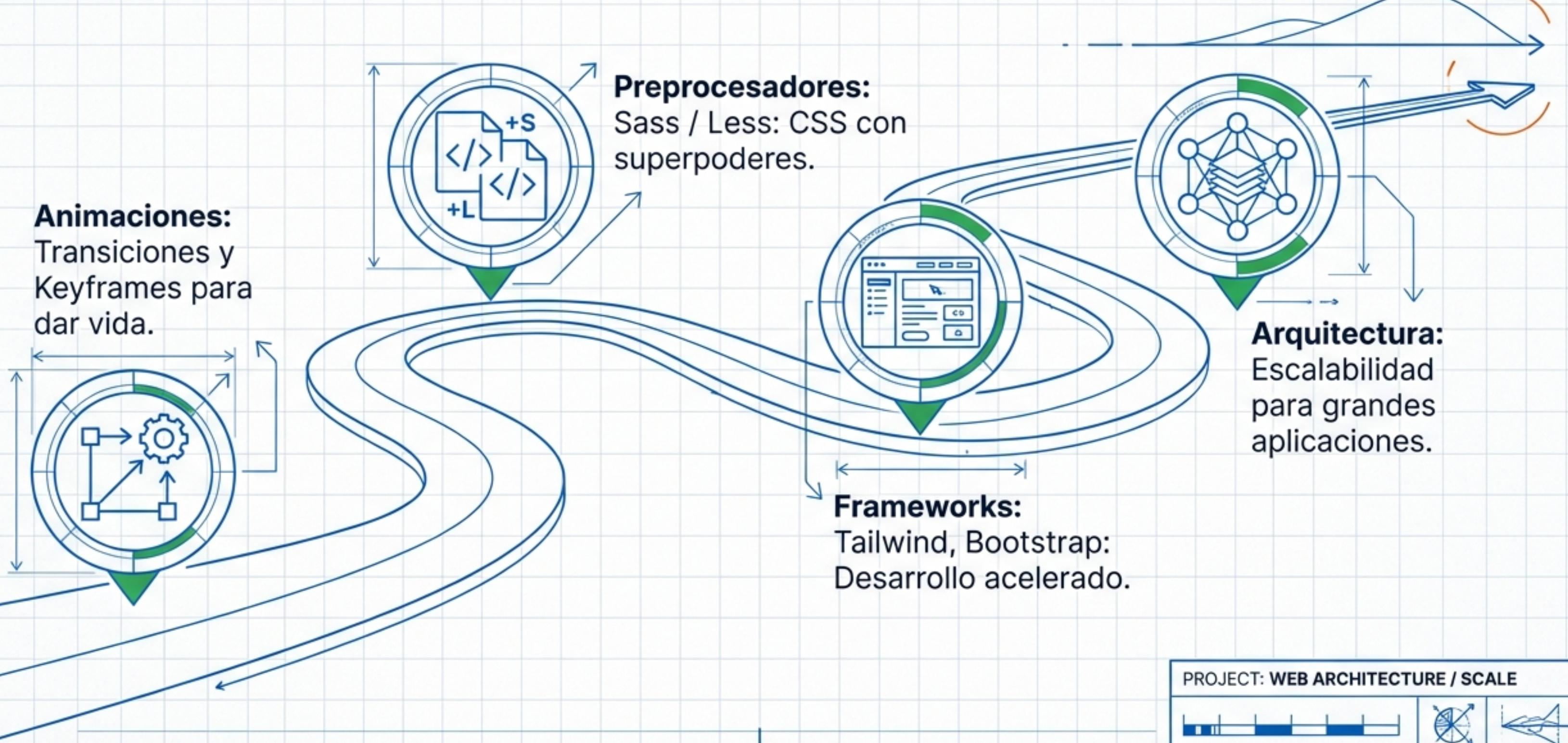
## DRY (Don't Repeat Yourself)

Reutiliza clases utilitarias para evitar código duplicado.

“El código se lee más veces de las que se escribe.”



# 10. ¿Qué sigue después de CSS?



PROJECT: WEB ARCHITECTURE / SCALE



# Conclusión: El Arquitecto Digital

**Un desarrollador Junior no es quien “hace que algo se vea bonito”, sino quien construye estilos escalables, mantenibles y coherentes con una arquitectura sólida.**

**Practica constantemente. ➔  
Rompe layouts. Construye el futuro.**



PROJECT: WEB ARCHITECTURE / SCALE